WHAT IS CLAIMED IS:

1   1.   A method for accessing data in a distributed database environment,

2   comprising:

3   receiving, with a client program, multiple requests for data from a database object

4   satisfying specified search predicates from an application program, wherein each request

5   includes a request for at least one row from the database object;

6   transferring, with the client program, a database command and a rowset parameter

7   indicating a maximum number of rows to return to a server program over a network if the

8   requested row is not maintained by the client program;

9   generating a data block with the server program including rows from the database

10   object satisfying the search predicates in response to the database command, wherein the rows

11   included in the data block do not exceed the rowset parameter;

12   transferring, with the server program, the data block to the client program; and

13   returning, with the client program, at least one requested row from the received data block in

14   response to one request for the at least one row of data from the application program.

1   2.   The method of claim 1, wherein the multiple requests specify orientation

2   information for a row from the database object satisfying the specified search predicates.

1   3.   The method of claim 1, wherein the server program maintains a maximum block

2   size parameter, and wherein the data block is further generated to not exceed the block size

3   parameter.

1   4.   The method of claim 1, wherein the multiple requests received by the client

2   program from the application program comprise single-row fetch requests and the rows from

3   the database object that satisfy the search predicates are returned as part of a scrollable cursor

4   created by the application program.

1   5.   The method of claim 4, further comprising:

2   maintaining, with the client program, a first pointer addressing a last accessed row from

3   the received data block and a client cursor addressing a last requested row from the scrollable

4   cursor; and

5   incrementing the client cursor to an entry in the database object corresponding to the

6   last row returned to the application program.

1   6.   The method of claim 5, wherein the server program maintains a server cursor

2   addressing the last row from the database object included in a last data block returned to the

3   client program.

1   7.   The method of claim 6, wherein the client program manages the client cursor to

2   ensure that the correct row is returned from the server in order to satisfy the client requests and

3   wherein the client program is capable of sending a command to the server program to correct

4   the server cursor position.

1   8.   The method of claim 1, wherein the client program and server program

2   communicate using the Distributed Relational Database Architecture (DRDA).

1   9.   The method of claim 1, wherein the search predicates are defined with a

2   database cursor that provides a result table subset of the database object that satisfies the

3   search predicates.

1   10.   The method of claim 8, wherein the database command transferred by the client

2   program comprises an open cursor command.

1   11.   The method of claim 8, wherein the database command transferred by the client

2   program comprises a continued request for rows from an open cursor.

1   12.   The method of claim 1, further comprising:

2   determining, with the client program, whether the data block includes less rows than the

3   rowset parameter;

4   determining, with the client program, a difference between the rowset parameter and a

5   number of rows included in the data block if the data block includes less rows than the rowset

6   parameter; and

7   sending, with the client program, a command to the server program to transmit the

8   difference of rows.

1   13.   A method for accessing data in a distributed database environment,

2   comprising:

3   receiving, with a client program, multiple requests for at least one row of data from a

4   database object satisfying specified search predicates from an application program, wherein

5   each request includes a request for at least one row from the database object satisfying the

6   specified search predicates;

7   transferring, with the client program, a database command and a rowset parameter

8   indicating a maximum number of rows to return to a first server program over a network if the

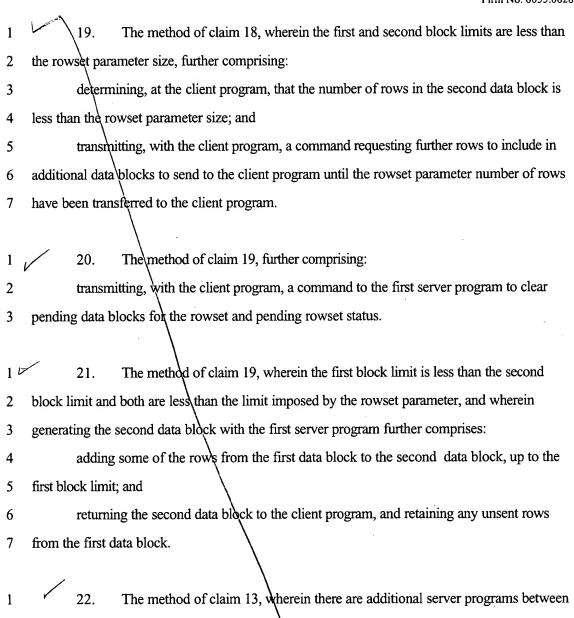9   requested row is not maintained by the client program;

10   transferring, with the first server program, a database command and the rowset

11   parameter to a second server program over the network if the requested row is not maintained

12   by the first server program;

13   generating a first data block with the second server program including rows from the

14   database object satisfying the search predicates in response to the database command, wherein

15   the rows included in the first data block do not exceed the rowset parameter;

16   transferring, with the first server program, the first data block to the first server

17   program;

18   generating a second data block with the first server program including rows from the

19   first data block, wherein the rows in the second data block do not exceed the rowset

20   parameter;

21   transferring, with the first server program, the second data block to the client program;

22   and

23   returning, with the client program, at least one requested row from the received data

24   block in response to one request for the at least one row of data from the application program.

1   14.   The method of claim 13, wherein the first server program maintains a block

2   limit, wherein a number of rows the first server program includes in the second data block does

3   not exceed the block limit.

1   15.   The method of claim 14, wherein generating the second data block with the first

2   server program from the rows in the first data block comprises:

3   adding rows from the first data block to the second data block until a size of the second

4   data block reaches one of the rowset parameter or the block limit; and

5   buffering the rows in the first data block that are not added to the second data block.

1 16. The method of claim 13, wherein the secon...
2 limit, wherein a number of rows the second server program inclu... ...m maintains a block
3 further does not exceed the block limit. ...data block

1 17. The method of claim 13, wherein the first server program maintains a first blo...
2 limit and wherein the second data block further does not exceed the first block limit and
3 wherein the second server program maintains a second block limit, wherein the first data block
4 further does not exceed the second block limit.

1 18. The method of claim 17, wherein the first block limit is greater than the second
2 block limit and both are less than the limit imposed by the rowset parameter and wherein
3 generating the second data block with the first server program from the rows in the first data
4 block comprises:

5 adding all the rows from the first data block to the second data block, wherein the rows
6 added to the second data block is less than the rowset parameter;
7 transmitting, with the first server program, a database command to the second server
8 program requesting a shortfall of rows equal to the rowset parameter minus the number of rows
9 added to the second data block; and
10 receiving, with the first server program, a third data block from the second server
11 program including the shortfall of rows;
12 adding, with the first server program, rows from the third data block, up to the first
13 block limit, to the pending second data block;
14 repeating the sending of a database command to the second server program and the
15 receiving of additional rows until the first block limit is satisfied; and
16 returning the second data block to the client program.

1   19.   The method of claim 18, wherein the first and second block limits are less than

2   the rowset parameter size, further comprising:

3   determining, at the client program, that the number of rows in the second data block is

4   less than the rowset parameter size; and

5   transmitting, with the client program, a command requesting further rows to include in

6   additional data blocks to send to the client program until the rowset parameter number of rows

7   have been transferred to the client program.

1   20.   The method of claim 19, further comprising:

2   transmitting, with the client program, a command to the first server program to clear

3   pending data blocks for the rowset and pending rowset status.

1   21.   The method of claim 19, wherein the first block limit is less than the second

2   block limit and both are less than the limit imposed by the rowset parameter, and wherein

3   generating the second data block with the first server program further comprises:

4   adding some of the rows from the first data block to the second data block, up to the

5   first block limit; and

6   returning the second data block to the client program, and retaining any unsent rows

7   from the first data block.

1   22.   The method of claim 13, wherein there are additional server programs between

2   the first server program and second server program through which the rows from the database

3   object are transferred.

1   23.   A system for accessing data in a distributed database environment, comprising:

2   a computer database including at least one database object;

3   a client computer;

4   a server computer;

5   a network enabling communication between the client computer and server computer;

6   a client program executed by the client computer to perform:

7   (i) receiving multiple requests for data from a database object satisfying

8   specified search predicates from an application program, wherein each request includes

9   a request for at least one row from the database object;

10   (ii) transferring a database command and a rowset parameter indicating a

11   maximum number of rows to return to the server program over a network if the

12   requested row is not maintained by the client program;

13   a server program executed by the client computer to perform:

14   (i) generating a data block including rows from the database object satisfying

15   the search predicates in response to the database command from the client program,

16   wherein the rows included in the data block do not exceed the rowset parameter;

17   (ii) transferring the data block to the client program, wherein the client program

18   returns at least one requested row from the received data block in response to one

19   request for the at least one row of data from the application program.

1   24.   The system of claim 23, wherein the multiple requests specify orientation

2   information for a row from the database object satisfying the specified search predicates.

1   25.   The system of claim 23, wherein the server program maintains a maximum

2   block size parameter, and wherein the data block is further generated to not exceed the block

3   size parameter.

1   26.   The system of claim 23, wherein the multiple requests received by the client

2   program from the application program comprise single-row fetch requests and the rows from

3   the database object that satisfy the search predicates are returned as part of a scrollable cursor

4   created by the application program.

1   27.   The system of claim 26, wherein the client program further performs:

2   maintaining a first pointer addressing a last accessed row from the received data block

3   and a client cursor addressing a last requested row from the scrollable cursor; and

4   incrementing the client cursor to an entry in the database object corresponding to the

5   last row returned to the application program.

1   28.   The system of claim 27, wherein the server program maintains a server cursor

2   addressing the last row from the database object included in a last data block returned to the

3   client program.

1   29.   The system of claim 28, wherein the client program manages the client cursor to

2   ensure that the correct row is returned from the server in order to satisfy the client requests and

3   wherein the client program is capable of sending a command to the server program to correct

4   the server cursor position.

1   30.   The system of claim 23, wherein the client program and server program

2   communicate using the Distributed Relational Database Architecture (DRDA).

1   31.   The system of claim 23, wherein the search predicates are defined with a

2   database cursor that provides a result table subset of the database object that satisfies the

3   search predicates.

1   32.   The system of claim 31, wherein the database command transferred by the

2 client program comprises an open cursor command.

1   33.   The system of claim 31, wherein the database command transferred by the

2 client program comprises a continued request for rows from an open cursor.

1   34.   The system of claim 23, wherein the client program further performs:

2   determining whether the data block includes less rows than the rowset parameter;

3   determining a difference between the rowset parameter and a number of rows included

4 in the data block if the data block includes less rows than the rowset parameter; and

5   sending a command to the server program to transmit the difference of rows.

1   35.   A system for accessing data in a distributed database environment, comprising:

2   a distributed computing environment including at least one client program, first server

3 program, second server program, and at least one database object;

4   a network enabling communication among the client program, first server program, and

5 second server program;

6   means, performed by the client program, for receiving multiple requests for at least one

7 row of data from a database object satisfying specified search predicates from an application

8 program, wherein each request includes a request for at least one row from the database object

9 satisfying the specified search predicates;

10   means, performed by the client program, for transferring a database command and a

11 rowset parameter indicating a maximum number of rows to return to the first server program

12 over the network if the requested row is not maintained by the client program;

13   means, performed by the first server program, for transferring a database command and

14   the rowset parameter to the second server over the network if the requested row is not

15   maintained by the first server program;

16   means, performed by the second server program, for generating a first data block

17   including rows from the database object satisfying the search predicates in response to the

18   database command from the first server program, wherein the rows included in the first data

19   block do not exceed the rowset parameter;

20   means, performed by the first server program, for transferring the first data block to the

21   first server program;

22   means, performed by the first server program, for generating a second data block

23   including rows from the first data block, wherein the rows in the second data block do not

24   exceed the rowset parameter;

25   means, performed by the first server program, for transferring the second data block to

26   the client program; and

27   means, performed by the client program, for returning at least one requested row from

28   the received data block in response to one request for the at least one row of data from the

29   application program.

1   /4   36.   The system of claim 35, wherein the first server program maintains a block limit,

2   wherein a number of rows the first server program includes in the second data block does not

3   exceed the block limit.

4

15

1   37.   The system of claim 36, wherein the means for generating the second data

2   block with the first server program from the rows in the first data block further performs:

3   adding rows from the first data block to the second data block until a size of the second

4   data block reaches one of the rowset parameter or the block limit; and

5   buffering the rows in the first data block that are not added to the second data block.

16

1   38.   The system of claim 35, wherein the second server program maintains a block

2   limit, wherein a number of rows the second server program includes in the first data block

3   further does not exceed the block limit.

17

1   39.   The system of claim 35, wherein the first server program maintains a first block

2   limit and wherein the second data block further does not exceed the first block limit and

3   wherein the second server program maintains a second block limit, wherein the first data block

4   further does not exceed the second block limit.

18

1   40.   The system of claim 39, wherein the first block limit is greater than the second

2   block limit and both are less than the limit imposed by the rowset parameter, and wherein the

3   means for generating the second data block with the first server program from the rows in the

4   first data block performs:

5   adding all the rows from the first data block to the second data block, wherein the rows

6   added to the second data block is less than the rowset parameter;

7   transmitting, with the first server program, a database command to the second server

8   program requesting a shortfall of rows equal to the rowset parameter minus the number of rows

9   added to the second data block;

10   receiving, with the first server program, a third data block from the second server

11   program including the shortfall of rows;

12   adding rows from the third data block, up to the first block limit, to the pending second

13   data block;

14   repeating the sending of a database command to the second server program and the

15   receiving of additional rows until the first block limit is satisfied; and

16   returning the second data block to the client program.

1   19   41.   The system of claim 40, wherein the first and second block limits are less than

2   the rowset parameter size, further comprising:

3   means, performed by the client program, for determining whether the number of rows in

4   the second data block is less than the rowset parameter size; and

5   means, performed by the first client program, for transmitting a command requesting

6   further rows to include in additional data blocks to send to the client program until the rowset

7   parameter number of rows have been transferred to the client program.

1   20   42.   The system of claim 41, further comprising:

2   means, performed by the client program, for transmitting a command to the first server

3   program to clear pending data blocks for the rowset and pending rowset status.

1   21   43.   The system of claim 41, wherein the first block limit is less than the second

2   block limit and both are less than the limit imposed by the rowset parameter, and wherein the

3   means for generating the second data block with the first server program further performs:

4   adding some of the rows from the first data block to the second data block, up to the

5   first block limit; and

6   returning the second data block to the client program, and retaining any unsent rows

7   from the first data block.

1   44.   The system of claim 35, further comprising additional server programs between

2   the first server program and second server program through which the rows from the database

3   object are transferred.

1   45.   Multiple computer readable media including instructions in a client program and

2   server program that communicate over a network to cause computers to enable access to data

3   in a distributed database environment by:

4   receiving, with the client program, multiple requests for data from a database object

5   satisfying specified search predicates from an application program, wherein each request

6   includes a request for at least one row from the database object;

7   transferring, with the client program, a database command and a rowset parameter

8   indicating a maximum number of rows to return to the server program over a network if the

9   requested row is not maintained by the client program;

10   generating a data block with the server program including rows from the database

11   object satisfying the search predicates in response to the database command, wherein the rows

12   included in the data block do not exceed the rowset parameter;

13   transferring, with the server program, the data block to the client program; and

14   returning, with the client program, at least one requested row from the received data block in

15   response to one request for the at least one row of data from the application program.

1   46.   The multiple computer readable media of claim 45, wherein the multiple

2   requests specify orientation information for a row from the database object satisfying the

3   specified search predicates.

1   47.   The multiple computer readable media of claim 45, wherein the server program

2   maintains a maximum block size parameter, and wherein the data block is further generated to

3   not exceed the block size parameter.

1   48.   The multiple computer readable media of claim 45, wherein the multiple

2   requests received by the client program from the application program comprise single-row fetch

3   requests and the rows from the database object that satisfy the search predicates are returned

4   as part of a scrollable cursor created by the application program.

1   49.   The multiple computer readable media of claim 48, further comprising:

2   maintaining, with the client program, a first pointer addressing a last accessed row from

3   the received data block and a client cursor addressing a last requested row from the scrollable

4   cursor; and

5   incrementing the client cursor to an entry in the database object corresponding to the

6   last row returned to the application program.

1   50   The multiple computer readable media of claim 49, wherein the server program

2   maintains a server cursor addressing the last row from the database object included in a last

3   data block returned to the client program.

1   51.   The multiple computer readable media of claim 50, wherein the client program

2   manages the client cursor to ensure that the correct row is returned from the server in order to

3   satisfy the client requests and wherein the client program is capable of sending a command to

4   the server program to correct the server cursor position.

52.   The multiple computer readable media of claim 45, wherein the client program

and server program communicate using the Distributed Relational Database Architecture

(DRDA).

53.   The multiple computer readable media of claim 45, wherein the search

predicates are defined with a database cursor that provides a result table subset of the database

object that satisfies the search predicates.

54.   The multiple computer readable media of claim 52, wherein the database

command transferred by the client program comprises an open cursor command.

55.   The multiple computer readable media of claim 45, further comprising

       determining, with the client program, whether the data block includes less rows than the

rowset parameter;

       determining, with the client program, a difference between the rowset parameter and a

number of rows included in the data block if the data block includes less rows than the rowset

parameter; and

       sending, with the client program, a command to the server program to transmit the

difference of rows.

56.   Multiple computer readable media including instructions in a client program,

first server program, and second server program that communicate over a network to cause

computers to enable access to data in a network distributed database environment by:

       receiving, with the client program, multiple requests for at least one row of data from a

database object satisfying specified search predicates from an application program, wherein

each request includes a request for at least one row from the database object satisfying the

specified search predicates;

8   transferring, with the client program, a database command and a rowset parameter

9   indicating a maximum number of rows to return to the first server program over a network if the

10   requested row is not maintained by the client program;

11   transferring, with the first server program, a database command and the rowset

12   parameter to a second server program over the network if the requested row is not maintained

13   by the first server program;

14   generating a first data block with the second server program including rows from the

15   database object satisfying the search predicates in response to the database command, wherein

16   the rows included in the first data block do not exceed the rowset parameter;

17   transferring, with the first server program, the first data block to the first server

18   program;

19   generating a second data block with the first server program including rows from the

20   first data block, wherein the rows in the second data block do not exceed the rowset

21   parameter;

22   transferring, with the first server program, the second data block to the client program;

23   and

24   returning, with the client program, at least one requested row from the received data

25   block in response to one request for the at least one row of data from the application program.

1   57.   The multiple computer readable media of claim 56, wherein the first server

2   program maintains a block limit, wherein a number of rows the first server program includes in

3   the second data block does not exceed the block limit.

1   58.   The multiple computer readable media of claim 57, wherein generating the

2   second data block with the first server program from the rows in the first data block comprises:

3   adding rows from the first data block to the second data block until a size of the second

4   data block reaches one of the rowset parameter or the block limit; and

5   buffering the rows in the first data block that are not added to the second data block.

1   59.   The multiple computer readable media of claim 56, wherein the second server

2   program maintains a block limit, wherein a number of rows the second server program includes

3   in the first data block further does not exceed the block limit.

1   60.   The multiple computer readable media of claim 56, wherein the first server

2   program maintains a first block limit and wherein the second data block further does not exceed

3   the first block limit and wherein the second server program maintains a second block limit,

4   wherein the first data block further does not exceed the second block limit.

1   61.   The multiple computer readable media of claim 60, wherein the first block limit

2   is greater than the second block limit and both are less than the limit imposed by the rowset

3   parameter, and wherein generating the second data block with the first server program from the

4   rows in the first data block comprises:

5   adding all the rows from the first data block to the second data block, wherein the rows

6   added to the second data block is less than the rowset parameter;

7   transmitting, with the first server program, a database command to the second server

8   program requesting a shortfall of rows equal to the rowset parameter minus the number of rows

9   added to the second data block;

10   receiving, with the first server program, a third data block from the second server

11   program including the shortfall of rows;

12   adding, with the first server program, rows from the third data block, up to the first

13   block limit, to the pending second data block;

14   repeating the sending of a database command to the second server program and the

15   receiving of additional rows until the first block limit is satisfied; and

16   returning the second data block to the client program.

1   62.   The multiple computer readable media of claim 55, wherein the first and second

2   block limits are less than the rowset parameter size, further comprising:

3   determining, at the client program, that the number of rows in the second data block is

4   less than the rowset parameter size; and

5   transmitting, with the client program, a command requesting further rows to include in

6   additional data blocks to send to the client program until the rowset parameter number of rows

7   have been transferred to the client program.

1   63.   The multiple computer readable media of claim 62, further comprising:

2   transmitting, with the client program, a command to the first server program to clear

3   pending data blocks for the rowset and pending rowset status.

1   64.   The multiple computer readable media of claim 62, wherein the first block limit

2   is less than the second block limit and both are less than the limit imposed by the rowset

3   parameter, and wherein generating the second data block with the first server program further

4   comprises:

5   adding some of the rows from the first data block to the second data block, up to the

6   first block limit; and

7   returning the second data block to the client program, and retaining any unsent rows

8   from the first data block.